

Weekly Report(2019.1.14-2019.1.20)

DONE

1. 学习 Graph Embedding
2. 阅读专业书籍
3. 整理size实验的代码

小结

- 本周学习了Graph Embedding的相关概念，看了吴军的《浪潮之巅》，了解了计算机领域的发展历史。整理了之前的JS代码。
- 下一周是实验室放假前的最后一周了，抓紧机会再多学习一些。

学习记录

学习日期	学习事项	学习时间
周一	配置VS Code，体验插件，整理	5h
周二	学习graph embedding	5h
周三	学习graph embedding	3h
周四	整理代码	6h
周五	阅读《浪潮之巅》	9h
周六	休息	0h
周日	休息	0h

PLAN

短期计划

1. 学习Graph Embedding。

中期计划

1. 自学less、webpack、promise、\$.ajax等等。
2. 能阅读购买的ES6、JavaScript标准等书的一些部分。

长期计划

1. 推进VIS 2019的项目进度。
2. 使用所学的网页工具实现自己的idea。

APPENDIX

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6.     <title>violin chart</title>
7.     <meta name="viewport" content="width=device-width, initial-
scale=1">
8.     <style>
9.         body {font-family: sans-serif, 微软雅黑; font-size: 12px;}
10.        /* title */
11.        .title {font-size: 20px; text-anchor: middle;}
12.        /* box plot */
13.        line {stroke: #000; stroke-width: 2px;}
14.        .box {fill: rgb(68, 147, 212);}
15.        .center {stroke-dasharray: 3,3;}
16.        .median {stroke: #000; stroke-width: 4px;}
17.        .whisker {stroke: #000; stroke-width: 4px;}
18.        .outlier {fill: #fff; stroke: gray; stroke-width: 3px;}
19.        .boxText {font-size: 12px;}
20.        /* violin */
21.        .area {fill: rgba(101, 184, 252, 0.3);}
22.        /* axis */
23.        .axis path,
24.        .axis line {fill:none; stroke: #000; shape-rendering: crispEdge
s;}
25.        .axis text {font-size: 12px;}
26.    </style>
27.    <script src="https://d3js.org/d3.v4.min.js"></script>
28. </head>
29. <body>
30.     <script>
31.         function wj_draw(wjData){
32.             // svg
33.             var svgWidth = 400;
34.             var svgHeight = 400;
35.             var padding = {left: 60, right: 60, top: 60, bottom: 60};
36.             var width = svgWidth - padding.left - padding.right;
37.             var height = svgHeight - padding.top - padding.bottom;
38.             // other para
39.             var axisLeftPadding = -20;
40.             var axisBottomPadding = 20;
41.             var box_padding = width*0.4;
42.             // add svg
43.             var svg = d3.select('body')
44.                 .append('svg')
45.                 .attr('width', svgWidth)
46.                 .attr('height', svgHeight);
47.             // add title
```

```

48.     var titleText = wjData.class + " 列" + wjData.num + " violi
    n chart";
49.     var title = svg.append("g")
50.         .append("text")
51.         .attr("class", "title")
52.         .text(titleText)
53.         .attr("x", svgWidth / 2)
54.         .attr("y", 20);
55.
56.         // data transform
57.         // dataSet as 12345 -> 3 2 8 5 1
58.         var dataSet = wjData.dataset;
59.         // violinData as 1 5 8 2 3
60.         var violinData = [].concat(dataSet);
61.         violinData.reverse();
62.         // boxData as 1 1 1 2 2 3 ...
63.         var boxData = [];
64.         for(var i=0; i<dataSet.length; i++){
65.             for(var num=0; num<dataSet[i]; num++){
66.                 boxData.push(i+1);
67.             }
68.         }
69.         // draw two chart
70.         box_plot(boxData);
71.         violin_chart(violinData);
72.
73.         /*
74.             the function for draw
75.         */
76.
77.         function violin_chart(data){
78.             console.log("draw the violin chart");
79.             console.log(data);
80.             // scale
81.             var xScale = d3.scaleLinear()
82.                 .domain([0, data.length - 1])
83.                 .range([0, height]);
84.             var yScale = d3.scaleLinear()
85.                 .domain([0, d3.max(data)])
86.                 .range([width/2, 0]);
87.             // area as reverse the x and y
88.             var violinLeft = d3.area()
89.                 .y( (d, i) => xScale(i))
90.                 .x0( width/2, 0)
91.                 .x1( (d, i) => yScale(d))
92.                 .curve( d3.curveCardinal);
93.             // .curve( d3.curveNatural);
94.             // from https://github.com/xswei/d3
            -shape/blob/master/README.md#areas
95.             var violinRight = d3.area()
96.                 .y( (d, i) => xScale(i))
97.                 .x0( width/2, 0)

```

```

98.             .x1( (d, i) => width - yScale(d))
99.             .curve( d3.curveCardinal);
100.            // .curve( d3.curveNatural);
101.            // add group
102.            var g = svg.append("g")
103.                .attr("transform", "translate(" + padding.l
left + "," + padding.top + ")");
104.            // add area
105.            var pathLeft = g.append("path")
106.                .attr("class", "area")
107.                .attr("d", violinLeft(data));
108.            var pathRight = g.append("path")
109.                .attr("class", "area")
110.                .attr("d", violinRight(data));
111.
112.            // add axis as reverse the x and y
113.            var gX = svg.append("g")
114.                .attr("transform", "translate(" + (padding.
left + axisLeftPadding) + "," + padding.top + ")");
115.            var xAxis = gX.append("g")
116.                .attr("class", "axis")
117.                .call(d3.axisLeft(xScale).tickFormat(
(d, i) => data.length-d ));
118.            // another axis
119.            var gY = svg.append("g")// a little far
120.                .attr("transform", "translate(" + padding.l
left + "," + (padding.top + axisBottomPadding) + ")");
121.
122.            var yAxis = gY.append("g")
123.                .attr("class", "axis")
124.                .attr("transform", "translate(0," + hei
ght + ")")
125.                .call(d3.axisBottom(yScale).tickFormat(
function(d, i){
126.                    if( i%2 == 0) return d;
127.                    else return "";
128.                } ));
129.            var dataMax = d3.max(data);
130.            var yAxisReverse = gY.append("g")
131.                .attr("class", "axis")
132.                .attr("transform", "translate(" + width
h/2 + "," + height + ")")
133.                .call(d3.axisBottom(yScale).tickFormat(
function(d, i){
134.                    if( i%2 == 0){
135.                        return dataMax-d;
136.                    }
137.                    else return "";
138.                } ));
139.            function box_plot(data){
140.                console.log("draw the box plot");

```

```

141.         console.log(data);
142.         // cal data index
143.         data.sort(d3.ascending);
144.         var n = data.length;
145.         var min = data[0];
146.         var max = data[n - 1];
147.         // quantile
148.         var quantileData = wj_quantile(data);
149.         // whisker index
150.         var whiskerIndices = wj_iqr(data, quantileData);
151.         // whisker data
152.         var whiskerData = [data[whiskerIndices[0]], data[whiske
rIndices[1]]];
153.         // outlier index
154.         var outlierIndices = d3.range(0, whiskerIndices[0])
155.             .concat(d3.range(whiskerIndices
[1] + 1, n));
156.         console.log("长度 " + n + "\n最小值 " + min + " 最大值 " +
max);
157.         console.log("下四分位数 " + quantileData[0] + " 中位数 " +
quantileData[1] + " 上四分位数 " + quantileData[2]);
158.         console.log("上下限index " + whiskerIndices + " 上下限 "
+ whiskerData + " 异常值的下标 " + outlierIndices);
159.         // scale
160.         var yScale = d3.scaleLinear()
161.             .domain([min, max])
162.             .range([height, 0]);
163.         // add group
164.         var g = svg.append("g")
165.             .attr("transform", "translate(" + padding.l
eft + "," + padding.top + ")");
166.         // add center line
167.         var centerLine = g.selectAll("line.center")
168.             .data([whiskerData]) // attention o
f [ ]
169.             .enter()
170.             .append('line', 'rect')
171.             .attr("class", "center")
172.             .attr("x1", width/2)
173.             .attr("y1", function(d, i){ return
yScale(d[0]);})
174.             .attr("x2", width/2)
175.             .attr("y2", (d, i) =>
yScale(d[1]));
176.         // box width
177.         // var box_padding = width*0.4;
178.         // add box
179.         var boxRect = g.selectAll("rect.box")
180.             .data([quantileData])
181.             .enter()
182.             .append("rect")
183.             .attr("class", "box")

```

```

184.         .attr("x", box_padding)
185.         .attr("y", (d, i) => yScale(d[2]))
186.         .attr("width", width - box_padding*2)
187.         .attr("height", (d, i) => yScale(d[0]) -
yScale(d[2]));
188.         // add median
189.         var medianLine = g.selectAll("line.median")
190.             .data([quantileData[1]])
191.             .enter()
192.             .append("line")
193.             .attr("class", "median")
194.             .attr("x1", box_padding)
195.             .attr("y1", (d, i) => yScale(d))
196.             .attr("x2", width - box_padding)
197.             .attr("y2", (d, i) => yScale(d));

198.         // add whisker
199.         var whiskerLine = g.selectAll("line.whisker")
200.             .data(whiskerData)
201.             .enter()
202.             .append("line", "circle", "text")
203.             .attr("class", "whisker")
204.             .attr("x1", box_padding)
205.             .attr("y1", (d, i) => yScale(d))
206.             .attr("x2", width - box_padding)
207.             .attr("y2", (d, i) => yScale(d));

208.         // add outlier
209.         var outlierCircle = g.selectAll("circle.outlier")
210.             .data(outlierIndices)
211.             .enter()
212.             .insert("circle", "text")
213.             .attr("class", "outlier")
214.             .attr("r", 5)
215.             .attr("cx", width / 2)
216.             .attr("cy", (inner_d, i) => ySc
ale(data[inner_d]));

217.         // index to data
218.         // 10 as 2.0 ; 100 as 2.00
219.         var tickFormat = yScale.tickFormat(10);
220.         // add box tick
221.         var boxTick = g.selectAll("text.boxText")
222.             .data(quantileData)
223.             .enter()
224.             .append("text")
225.             .attr("class", "boxText")
226.             .attr("dy", 0)
227.             .attr("dx", function(d, i) { return i &
1 ? 20 : -20 })
228.             .attr("x", function(d, i) { return i &
1 ? width - box_padding : box_padding })
229.             .attr("y", (d, i) => yScale(d))

```

```

230.         .attr("text-anchor", function(d, i) { r
return i & 1 ? "start" : "end" })
231.         .text(tickFormat);
232.         // add whisker tick
233.         var whiskerTick = g.selectAll("text.whisker")
234.             .data(whiskerData)
235.             .enter()
236.             .append("text")
237.             .attr("class", "boxText")
238.             .attr("dy", 0)
239.             .attr("dx", 20)
240.             .attr("x", width - box_padding)
241.             .attr("y", (d, i) => yScale(d))
242.             .text(tickFormat);
243.         // function for box plot
244.         function wj_iqr(d, quantileData){
245.             // quantile
246.             var q1 = quantileData[0];
247.             var q3 = quantileData[2];
248.             var iqr = (q3 - q1 ) * 1.5;
249.             var i = -1, j = d.length;
250.             // find the whirsker
251.             while( d[++i] < q1 - iqr);
252.             while( d[--j] > q3 + iqr);
253.             return [i, j];
254.         }
255.         function wj_quartile(d){
256.             // use d3 quantile
257.             var q1 = d3.quantile(d, 0.25);
258.             var q2 = d3.quantile(d, 0.5);
259.             var q3 = d3.quantile(d, 0.75);
260.             return [q1, q2, q3];
261.         }
262.     }
263. }
264. </script>
265. <script>
266.     var wjData = [
267.         {"class": "density", "num": "1", "dataset": [3, 2, 8, 5, 1]},
268.         {"class": "density", "num": "3", "dataset": [4, 3, 6, 2, 4]},
269.         {"class": "density", "num": "5", "dataset": [3, 5, 1, 4, 5]},
270.         {"class": "density", "num": "7", "dataset": [3, 2, 3, 6, 4]},
271.
272.         {"class": "correlation", "num": "1", "dataset": [4, 2, 10, 4,
273.         2]},
274.         {"class": "correlation", "num": "3", "dataset": [4, 3, 8, 2,
275.         5]},
276.         {"class": "correlation", "num": "5", "dataset": [5, 2, 11, 2,
277.         2]},
278.         {"class": "correlation", "num": "7", "dataset": [6, 2, 7, 5,
279.         1]},
280.         {"class": "cluster", "num": "1", "dataset": [1, 2, 4, 4, 0]},

```

```
276.         {"class": "cluster", "num": "3", "dataset": [0, 3, 5, 3, 0]},
277.         {"class": "cluster", "num": "5", "dataset": [1, 5, 3, 2, 0]},
278.         {"class": "cluster", "num": "7", "dataset": [4, 5, 1, 1, 0]}

279.     ];
280.     for(var i=0; i<wjData.length; i++){
281.         wj_draw(wjData[i]);
282.     }
283.     // wj_draw(wjData[0]);
284.     </script>
285. </body>
286. </html>
```